Inverse Planar Kinematics of a Two-Link Robot Arm Using Denavit – Hartenberg Representation

MECE 617 Activity

Ivan John A. Naparota
University of San Carlos – Talamban Campus
Department of Electrical and Electronics Engineering
ivanjohnnaparota66@gmail.com

***Abstract* — this activity presents the inverse kinematics of a two link planar robot arm using the Denavit – Hartenberg representation and exploits MATLAB's computing prowess to avoid manual solutions.**

Keywords* – *Inverse Kinematics, Denavit - Hartenberg, MATLAB

I. INTRODUCTION

In robotics, inverse kinematics makes use of the kinematics equations to determine the joint parameters that provide a desired position for each of the robot's end effecter [1]. Inverse kinematics could be solved with various methods but this activity only focuses on the application of Denavit – Hartenberg representation.

In Denavit – Hartenberg representation, it is necessary to identify and provide for the Denavit–Hartenberg parameters, it is the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator. Jacques Denavit and Richard Hartenberg introduced this convention in 1955 in order to standardize the coordinate frames for spatial linkages [2].

II. EQUATIONS AND MATLAB CODES

A. *DH Parameters and Transformation Matrices*

For a specific *i*, we obtain the T matrix, T = ^0A_i, which specifies the position and orientation of the endpoint of the manipulator with respect to the base coordinate system. Considering T matrix to be of the form [3]

$$T = \begin{bmatrix} cos\theta & -cos\alpha sin\theta & sin\alpha sin\theta & rcos\theta \\ sin\theta & cos\alpha cos\theta & -sin\alpha cos\theta & rsin\theta \\ 0 & sin\alpha & cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

See Figure 2,

Where:

d – depth along the previous joint's z axis

θ – angle about the previous z to align its x with the new origin

r – distance along the rotated x axis

α – rotation about the new x axis to put z in its desired orientation

These four parameters d, θ, r, and α are the so called DH parameters.

But, for this case, since we are only considering a planar two – link manipulator arm, values of α and d are equal to zero.

The homogeneous matrix ^0T_i which specifies the location of the *i*th coordinate frame with respect to the base coordinate system is the chain product of successive coordinate transformation matrices of $^{i-1}A_i$, and is expressed as [3]

$$^0T_i = {^0A_1}\,{^1A_2} \ldots {^{i-1}A_i} = \begin{bmatrix} 0R_i & 0P_i \\ 0 & 1 \end{bmatrix}$$

Where:

P – Position vector of the hand
$^0A_1$ – transformation matrix of Joint 1
$^1A_2$, – transformation matrix of Joint 2

Identifying $^0A_1$ and $^1A_2$,

Letting values of α and d to be zero and position our end effecter to a point (X, Y).

$$^0A_1 = \begin{bmatrix} cos\theta1 & -sin\theta1 & 0 & r1cos\theta1 \\ sin\theta1 & cos\theta1 & 0 & r1sin\theta1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$
{}^{1}A_2 = \begin{bmatrix} cos\theta 2 & -sin\theta 2 & 0 & r2cos\theta 2 \\ sin\theta 2 & cos\theta 2 & 0 & r2sin\theta 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Multiplication of the transformation matrices for each joint ${}^{0}T_2 = {}^{0}A_1\, {}^{1}A_2$ will give us a position vector of,

$$
0P_2 = \begin{bmatrix} r1 * cos\theta 1 - r2 * sin\theta 2 * sin\theta 1 + r2 * cos\theta 2 * cos\theta 1 \\ r1 * sin\theta 1 + r2 * cos\theta 2 * sin\theta 1 + r2 * sin\theta 2 * cos\theta 1 \end{bmatrix}
$$

Equating the X and Y of the end effecter to the x and y of the position vector, we get these equations

$$
X = r1 * cos\theta 1 - r2 * sin\theta 2 * sin\theta 1 + r2 * cos\theta 2 * cos\theta 1
$$

$$
Y = r1 * sin\theta 1 + r2 * cos\theta 2 * sin\theta 1 + r2 * sin\theta 2 * cos\theta 1
$$

This means that, for particular values of X, Y, r1, and r2, we can get the values of $\theta 1$ and $\theta 2$.

B. *MATLAB Codes*

```
clc;
clear all

syms t b

l1 = input('Link 1:');
l2 = input('Link 2:');
ex = input('End Effecter X:');
ey = input('End Effecter Y:');

T1 = [cos(t) -sin(t) 0 l1*cos(t)
sin(t) cos(t) 0 l1*sin(t)
0 0 1 0
0 0 0 1];

T2 = [cos(b) -sin(b) 0 l2*cos(b)
sin(b) cos(b) 0 l2*sin(b)
0 0 1 0
0 0 0 1];

T = T1*T2;

E1 = T(1,4) == ex;
E2 = T(2,4) == ey;

F = solve([E1,E2],[t,b]);

ans1 = [F.t(1,1) F.b(1,1)];

ee1 = eval(ans1);
EE1 = (ee1*180)/pi;
disp(EE1);

disp('OR');

ans2 = [F.t(2,1) F.b(2,1)];

ee2 = eval(ans2);
EE2 = (ee2*180)/pi;
disp(EE2);

disp('Type [TwoLinkInverseDH] in
Command Window to Input again :)')
```

IV. RESULTS AND DISCUSSIONS

As shown in Figure 3, the program will ask inputs, Length of Link 1, Length of Link 2, and the point location of the end-effecter x and y. Provided with the following inputs:

Length of Link 1: 3 units

Length of Link 2: 2 units

Position of End-Effecter X: 2

Position of End-Effecter Y: 2

We'll get two sets of angles as outcomes, (85.0011 and -114.6243) and (4.9989 and 114.6243), these are the joint variables that would define the end-effecter's position. These results are considered to be correct if we'll refer to the result of the previous activity of Inverse Kinematics using the Trigonometric approach (*refer to Figure 4*).

V. CONCLUSIONS

After presenting the data and the results, it is convincing that the Inverse Kinematics of a planar two link robot arm can be solved with Denavit – Hartenberg Representation, you just have to be able to identify the parameters of each joint necessary for this convention. Also, MATLAB's ability to solve numerical computations is of very much help for us to avoid manual solutions.

V. REFERENCES

1. Paul, Richard (1981). Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators. MIT Press, Cambridge, MA. ISBN 978-0-262-16082-7.
2. https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters

3. Fu, K. S., Gonzales, R. C., .Lee, C. S. G., ROBOTICS : Control, Sensing, Vision, and Intelligence

VI. DRAWINGS AND FIGURES

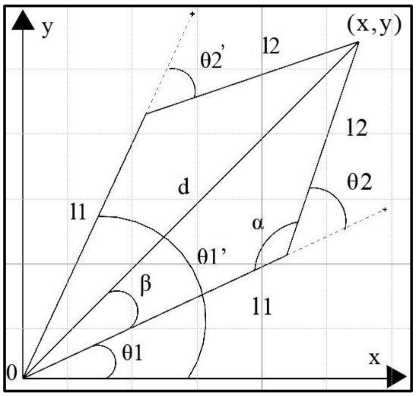
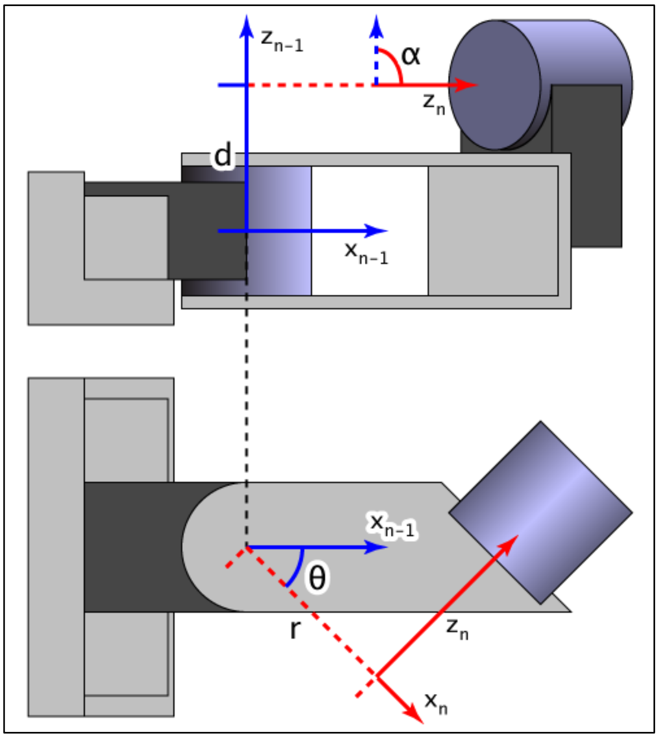


Figure 1.Planar Two - Link Robot Arm



https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters

Figure 2. DH Parameters

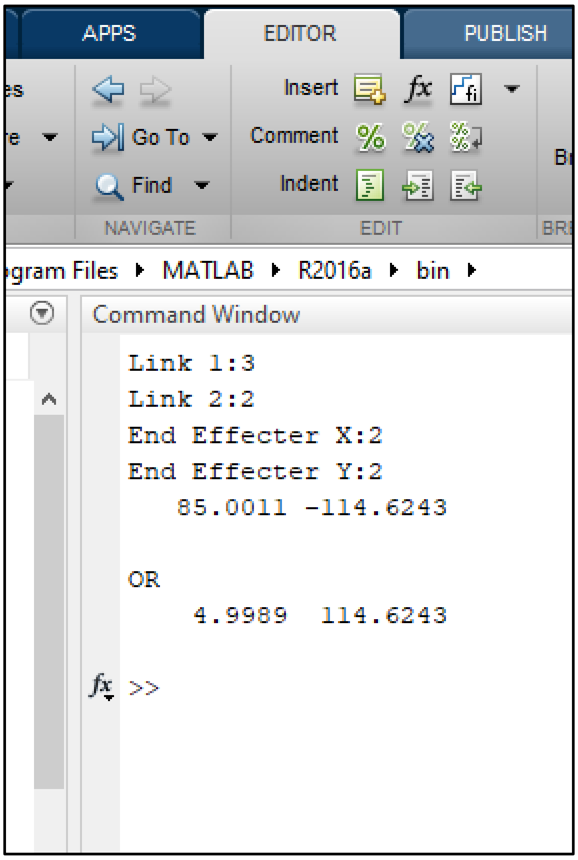


Figure 3.MATLAB Command Window

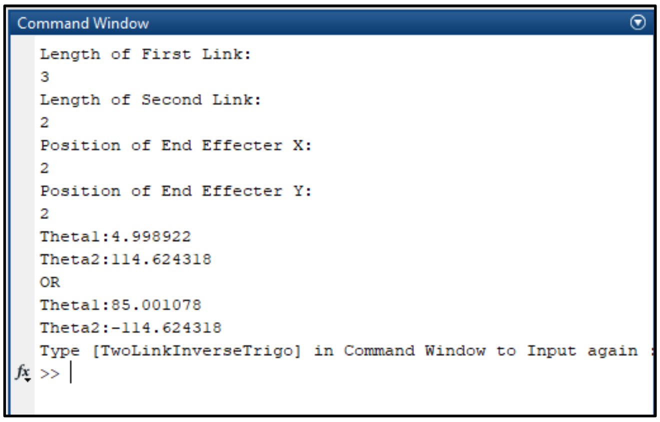


Figure 4.Inverse Kinematics Trigonometric Approach